

## SECTION I—CLAIMS

### **Amendment to the Claims:**

This listing of the claims will replace all prior versions and listings of claims in the application. Claims 1-46 are amended herein. No claims are canceled. No new claims are added. Claims 1-46 remain pending in the application.

### **Listing of Claims:**

1. (Currently amended) A method for processing a group of instructions, the method comprising: processing a group of instructions in topological dependence order;  
selecting an instruction to schedule, the instruction associated with an at least one operand,  
wherein the operand comprises a live range representing the operational duration of the  
operand relative to the group of instructions being processed;  
to schedule; allocating a register to one or more of the operands; determining if the live range of  
the operand spans a function call; if so,  
attempting to allocate a preserved register to the operand for its live range when its live range  
spans a function call; if not,  
attempting to allocate a scratch register to the operand for its live range when its live range does  
not span the function call; and  
if the determination is unknown, treating the live range as tentative; and if the allocation was  
successful,  
scheduling the instruction when the allocation of the preserved register or the scratch register is  
successful.

2. (Currently amended) The method of claim 1, further comprising:

~~wherein allocating a register to one or more of the operand includes: determining if the operand's use begins a live range or ends a live range; if the operand's use begins a live range, attempting to allocate a register;~~

~~terminating the attempt to allocate the preserved register or the scratch register to the operand for its live range when if the operand's use by the selected instruction falls at the end of the ends a live range; and [.]~~

~~marking the preserved register or the scratch register associated allocated to with the operand as available for reallocation.~~

3. (Currently amended) The method of claim 1, wherein attempting to allocate a ~~the~~ preserved register includes comprises:

~~determining if a preserved register is available for allocation from a list of previously used preserved registers; and if so,~~

~~allocating the available preserved register to the operand for its for the live range defined by the operand when the preserved register is in a list of previously used preserved registers marked as available for allocation.~~

4. (Currently amended) The method of claim 3, further including comprising:

~~, if a preserved register is not available from the list of previously used preserved registers,~~

~~attempting to find a tentative live range that includes a second preserved register that is in the list of previously used preserved registers and currently allocated to a tentative live range of a second operand; and if successful,~~

~~reallocating the second this preserved register for to the operand for its the live range defined by the operand; and reserving a different preserved register for to the second operand for its~~

the tentative live range when the attempt to find the second preserved register allocated to the second operand for its tentative live range is successful.

5. (Currently amended) The method of claim 4, further including comprising,  
, if the attempt to find a tentative live range that includes a preserved register that is in the list is not successful;

adding a new preserved register to the list of previously used preserved registers; and  
allocating this the new preserved register to the operand for its the live range defined by the  
operand when the attempt to find the second preserved register allocated to the second  
operand for its tentative live range fails.

6. (Currently amended) The method of claim 1, wherein attempting to allocate [[a]] the scratch register to the operand for its live range includes comprises:  
determining if a scratch register is available for allocation; if so, allocating a scratch register for  
the live range defined by the operand; if not,  
attempting to steal deallocate a second scratch register from a tentative live range of a second  
operand live range when the attempt to allocate the scratch register to the operand for its  
live range fails marked as tentative; determining if the attempted theft was successful; if  
so;

allocating the stolen second scratch register to the operand for its live range defined by the  
operand when the second scratch register is successfully deallocated from the second  
operand for its tentative live range; and if not,

attempting to allocate a the preserved register to the operand for its live range when the attempt  
to deallocate the second scratch register from the second operand for its tentative live  
range fails.

7. (Currently amended) The method of claim 1, further comprising:  
marking the live range of the operand as a tentative live range when current information about  
the live range is insufficient to determine whether or not the live range spans the function  
call;  
wherein treating the live range as tentative further includes: if both a scratch register and a  
preserved register are available for allocation, reserving  
pre-allocating both the scratch register and the preserved register to the operand for its tentative  
live range;  
processing one or more additional instructions allowing another instruction to be processed; and  
waiting for more additional information about the tentative live range.

8. (Currently amended) The method of claim 7, further comprising:  
including, when more information is available: if the last use of the live range has been  
scheduled,  
receiving the additional information about the tentative live range;  
allocating the reserved scratch register to the live range defined by the operand, and freeing  
deallocating the reserved preserved register from the operand for its tentative live range when the  
last use of the operand for its tentative live range has been scheduled; and  
if a function call has been scheduled before the last use of the live range or the reserved scratch  
register has been stolen, allocating the preserved register to the live range defined by the  
operand and freeing  
deallocating the reserved scratch register from the operand for its tentative live range when the  
function call is scheduled before the last use of the operand for its tentative live range or  
when the scratch register is reallocated by any other process.

9. (Currently amended) The method of claim 7, wherein further comprising:  
reserving a preserved register for the tentative live range includes: preferring to reserve  
attempting to allocate a currently unused preserved register that is in the list of previously used  
preserved registers to the operand for its tentative live range;[.] and before  
attempting to allocate reserving a new unused preserved register to the operand for the its  
tentative live range when the attempt to allocate the currently unused preserved register  
to the operand for its tentative live range fails.

10. (Currently amended) The method of claim [[9]] 7, further comprising:  
wherein treating the live range as tentative further includes: if a scratch register is not available  
for allocation,  
attempting to allocate a the preserved register to the operand for its tentative live range when the  
scratch register is determined to be unavailable.

11. (Currently amended) The method of claim 1, further comprising:  
wherein allocating a register to at least one or more of operands associated with the instruction  
includes: if no proper registers are available for allocation,  
inserting a register spill when the attempt to allocate the preserved register and the attempt to  
allocate the scratch register to the operand for its live range both fail; and  
either re-attempting to allocate the preserved register and the scratch register to the operand for  
its live range, a register, or selecting another a second instruction to schedule.

12. (Currently amended) An article of manufacture comprising[[.]] a machine readable medium  
having a plurality of machine accessible instructions logic thereon that, wherein when the  
instructions are executed, the instructions provide logic provides for processing a group  
of instructions, the logic further providing for:

processing a group of instructions in topological dependence order;

selecting an instruction to schedule, the instruction associated with an at least one operand,

wherein the operand comprises a live range representing the operational duration of the operand relative to the group of instructions being processed;

to schedule; allocating a register to one or more of the operands; determining if the live range of the operand spans a function call; if so,

attempting to allocate a preserved register to the operand for its live range when its live range spans a function call; if not,

attempting to allocate a scratch register to the operand for its live range when its live range does not span the function call; and

if the determination is unknown, treating the live range as tentative; and if the allocation was successful,

scheduling the instruction when the allocation of the preserved register or the scratch register is successful.

13. (Currently amended) The article of manufacture of claim 12, wherein the instructions providing for allocating a register to one or more of the operand includes the logic instructions further providing for:

determining if the operand's use begins a live range or ends a live range; if the operand's use begins a live range, attempting to allocate a register;

terminating the attempt to allocate the preserved register or the scratch register to the operand for its live range when if the operand's use by the selected instruction falls at the end of the ends a live range; and [[,]]

marking the preserved register or the scratch register associated allocated to with the operand as

available for reallocation.

14. (Currently amended) The article of manufacture of claim 12, wherein the logic instructions providing for attempting to allocate a the preserved register comprises includes instructions providing for:

determining if a preserved register is available for allocation from a list of previously used preserved registers; and if so,

allocating the available preserved register to the operand for its for the live range defined by the operand when the preserved register is in a list of previously used preserved registers marked as available for allocation.

15. (Currently amended) The article of manufacture of claim 14, the logic further including instructions providing for:

, if a preserved register is not available from the list of previously used preserved registers, attempting to find a tentative live range that includes a second preserved register that is in the list of previously used preserved registers and currently allocated to a tentative live range of a second operand; and if successful,

reallocating the second this preserved register for to the operand for its the live range defined by the operand; and reserving a different preserved register for to the second operand for its the tentative live range when the attempt to find the second preserved register allocated to the second operand for its tentative live range is successful.

16. (Currently amended) The article of manufacture of claim 15, further including the logic instructions further providing for:

, if the attempt to find a tentative live range that includes a preserved register that is in the list is not successful,

adding a new preserved register to the list of previously used preserved registers; and  
allocating this the new preserved register to the operand for its the live range defined by the  
operand when the attempt to find the second preserved register allocated to the second  
operand for its tentative live range fails.

17. (Currently amended) The article of manufacture of claim 12, wherein the logic instructions  
providing for attempting to allocate [[a]] the scratch register to the operand for its live  
range comprises:

includes instructions providing for: determining if a scratch register is available for allocation; if  
so, allocating a scratch register for the live range defined by the operand; if not,  
attempting to steal deallocate a second scratch register from a tentative live range of a second  
operand live range when the attempt to allocate the scratch register to the operand for its  
live range fails marked as tentative; determining if the attempted theft was successful; if  
so,

allocating the stolen second scratch register to the operand for its live range defined by the  
operand when the second scratch register is successfully deallocated from the second  
operand for its tentative live range; and if not,

attempting to allocate a the preserved register to the operand for its live range when the attempt  
to deallocate the second scratch register from the second operand for its tentative live  
range fails.

18. (Currently amended) The article of claim 12, wherein the logic instructions further providing  
for:  
marking treating the live range of the operand as a tentative live range when current information  
about the live range is insufficient to determine whether or not the live range spans the

function call;

further includes instructions providing for: if both a scratch register and a preserved register are available for allocation, reserving

pre-allocating both the scratch register and the preserved register to the operand for its tentative live range;

processing one or more additional instructions allowing another instruction to be processed; and waiting for more additional information about the tentative live range.

19. (Currently amended) The article of manufacture of claim 18, further including the logic instructions further providing for:

, when more information is available: if the last use of the live range has been scheduled, receiving the additional information about the tentative live range;

allocating the reserved scratch register to the live range defined by the operand, and freeing deallocated the reserved preserved register from the operand for its tentative live range when the

last use of the operand for its tentative live range has been scheduled; and

if a function call has been scheduled before the last use of the live range or the reserved scratch register has been stolen, allocating the preserved register to the live range defined by the operand and freeing

deallocating the reserved scratch register from the operand for its tentative live range when the function call is scheduled before the last use of the operand for its tentative live range or when the scratch register is reallocated by any other process.

20. (Currently amended) The article of manufacture of claim 18, wherein the logic instructions further providing for:

reserving a preserved register for the tentative live range includes instructions providing for:

preferring to reserve

attempting to allocate a currently unused preserved register that is in the list of previously used  
preserved registers to the operand for its tentative live range;[[],] and before  
attempting to allocate reserving a new unused preserved register to the operand for the its  
tentative live range when the attempt to allocate the currently unused preserved register  
to the operand for its tentative live range fails.

21. (Currently amended) The article of manufacture of claim [[20]] 18, wherein the logic  
instructions further providing for:  
treating the live range as tentative further includes instructions providing for: if a scratch register  
is not available for allocation,  
attempting to allocate a the preserved register to the operand for its tentative live range when the  
scratch register is determined to be unavailable.

22. (Currently amended) The article of manufacture of claim 12, wherein the logic instructions  
further providing for:  
allocating a register to at least one or more of operands associated with the instruction includes  
instructions providing for: if no proper registers are available for allocation,  
inserting a register spill when the attempt to allocate the preserved register and the attempt to  
allocate the scratch register to the operand for its live range both fail; and  
either re-attempting to allocate the preserved register and the scratch register to the operand for  
its live range, a register, or selecting another a second instruction to schedule.

23. (Currently amended) An apparatus comprising:  
an instruction scheduler to[[::]] process a group of instructions, the instruction scheduler to  
further in topological dependence order, and select an instruction to schedule, the

instruction associated with an at least one operand, to schedule the operand comprising a live range to represent the operational duration of the operand relative to the group of instructions to be processed; and

schedule an instruction, if the register allocation was successful; and

a register allocator to[[::]] allocate register to a live range associated with one or more operand,

determine if the live range of the operand spans a function call, if so, attempt to allocate a preserved register to the operand for its live range when its live range spans a function call[[::]] and if not, attempt to allocate a scratch register to the operand for its live range when its live range does not span the function call.

[[::]] and if the determination is unknown, treat the live range as tentative.

24. (Currently amended) The apparatus of claim 23, wherein the register allocator to further is capable of: determining if the operand's use begins a live range or ends a live range; if the operand's use begins a live range, attempting to allocate a register, terminate the attempt to allocate the preserved register or the scratch register to the operand for its live range when if the operand's use by the selected instruction is to fall at the end of the ends a live range, and mark marking the preserved register or the scratch register associated to be allocated to with the operand as available for reallocation.

25. (Currently amended) The apparatus of claim 23, wherein the register allocator to further is capable of, when attempting to allocate a preserved register: determining if a preserved register is available for allocation from a list of previously used preserved registers; and if so, allocating allocate the available preserved register to the operand for its for the live range defined by the

operand when the preserved register is in a list of previously used preserved registers to be marked as available for allocation.

26. (Currently amended) The apparatus of claim 25, wherein the register allocator to further is capable of, if a preserved register is not available from the list of previously used preserved registers: attempting

attempt to find a tentative live range that includes a second preserved register that is in the list of previously used preserved registers to be allocated to a tentative live range of a second operand.[]] and if successful, allocating

reallocate the second this preserved register for to the operand for its the live range, defined by the operand; and

reserve reserving a different preserved register for to the second operand for its the tentative live range when the attempt to find the second preserved register allocated to the second operand for its tentative live range is successful.

27. (Currently amended) The apparatus of claim 26, wherein the register allocator to further is capable of, if the attempt to find a tentative live range that includes a preserved register that is in the list is not successful: adding

add a new preserved register to the list of previously used preserved registers[]] and allocating this

allocate the new preserved register to the operand for its the live range defined by the operand when the attempt to find the second preserved register allocated to the second operand for its tentative live range fails.

28. (Currently amended) The apparatus of claim 23, wherein the register allocator is capable of, when attempting attempt to allocate [[a]] the scratch register to the operand for its live

range comprises the register allocator to further:

determining if a scratch register is available for allocation; if so, allocating a scratch register for the live range defined by the operand; if not, attempting

attempt to steal deallocate a second scratch register from a tentative live range of a second

operand live range when the attempt to allocate the scratch register to the operand for its live range fails, marked as tentative; determining if the attempted theft was successful; if so, allocating

allocate the stolen second scratch register to the operand for its live range defined by the operand when the second scratch register is successfully deallocated from the second operand for its tentative live range,[[;]] and if not, attempting

attempt to allocate [[a]] the preserved register to the operand for its live range when the attempt to deallocate the second scratch register from the second operand for its tentative live range fails.

29. (Currently amended) The apparatus of claim 23, wherein the register allocator to further is capable of, when treating a live range as tentative: if both a scratch register and a preserved register are available for allocation, reserving

pre-allocate both the scratch register and the preserved register to the operand for its tentative live range,[[;]]

process one or more additional instructions, allowing another instruction to be processed; and wait waiting for more additional information about the tentative live range.

30. (Currently amended) The apparatus of claim 29, wherein the register allocator to further is capable of, when more information is available: if the last use of the live range has been scheduled,

receive the additional information about the tentative live range, allocating the reserved scratch register to the live range defined by the operand, and freeing

deallocate the reserved preserved register from the operand for its tentative live range when the last use of the operand for its tentative live range has been scheduled,[[;]] and if a function call has been scheduled before the last use of the live range or the reserved scratch register has been stolen, allocating the preserved register to the live range defined by the operand and freeing

deallocate the reserved scratch register from the operand for its tentative live range when the function call is scheduled before the last use of the operand for its tentative live range or when the scratch register is reallocated by any other process.

31. (Currently amended) The apparatus of claim 30, wherein the register allocator to further is capable of, when reserving a preserved register for the tentative live range preferring to reserve attempt to allocate an [[a]] currently unused preserved register that is in the list of previously used preserved registers to the operand for its tentative live range, and before attempt to allocate reserving a new unused preserved register to the operand for the its tentative live range when the attempt to allocate the unused preserved register to the operand for its tentative live range fails.

32. (Currently amended) The apparatus of claim [[31]] 29, wherein the register allocator to further is capable of, when treating a live range as tentative: if a scratch register is not available for allocation, attempting attempt to allocate [[a]] the preserved register to the operand for its tentative live range when the scratch register is determined to be unavailable.

33. (Currently amended) The apparatus of claim 23, the register allocator to further  
is capable of: if no proper registers are available for allocation, inserting  
insert a register spill when the attempt to allocate the preserved register and the attempt to  
allocate the scratch register to the operand for its live range both fail,[[;]] and  
either re-attempting re-attempt to allocate the preserved register and the scratch register to the  
operand for its live range, a register, or selecting another select a second instruction to  
schedule.

34. (Currently amended) The apparatus of claim 23, wherein the register allocator further  
comprises includes:  
a scratch register allocator to allocate scratch registers to operands for their live ranges when the  
live ranges that do not span a function call;  
a preserved register allocator to allocate preserved registers to operands for their live ranges  
when the live ranges that span [[a]] the function call; and  
a tentative register allocator to allocate either [[a]] the scratch register or [[a]] the preserved  
register to their respective live ranges when it is unknown that are not immediately  
known whether or not the live ranges they span [[a]] the function call.

35. (Currently amended) A system comprising an integrated compiler to compile a group of  
instructions, wherein the integrated compiler comprises:  
a set of instructions to be compiled; a set of preserved registers capable of storing values that are  
to be preserved across function calls; a set of scratch registers capable of storing values  
that do not need to be preserved across function calls; and an integrated compiler having:  
an instruction scheduler to[[;]] process [[a]] the group of instructions, the instruction scheduler to  
further in topological dependence order, and select an instruction to schedule, the

instruction associated with an at least one operand, to schedule the operand comprising a live range to represent the operational duration of the operand relative to the group of instructions to be processed; and

schedule an instruction, if the register allocation was successful; and

a register allocator to[[::]] allocate register to a live range associated with one or more operand,

determine if the live range of the operand spans a function call, if so, attempt to allocate a preserved register to the operand for its live range when its live range spans a function call[[::]] and if not, attempt to allocate a scratch register to the operand for its live range when its live range does not span the function call.

[[::]] and if the determination is unknown, treat the live range as tentative.

36. (Currently amended) The system of claim 35, wherein the register allocator to further is capable of: determining if the operand's use begins a live range or ends a live range; if the operand's use begins a live range, attempting to allocate a register, terminate the attempt to allocate the preserved register or the scratch register to the operand for its live range when if the operand's use by the selected instruction is to fall at the end of the ends a live range, and

mark marking the preserved register or the scratch register associated to be allocated to with the operand as available for reallocation.

37. (Currently amended) The system of claim 35, wherein the register allocator to further is capable of, when attempting to allocate a preserved register: determining if a preserved register is available for allocation from a list of previously used preserved registers; and if so, allocating

allocate the available preserved register to the operand for its for the live range defined by the

operand when the preserved register is in a list of previously used preserved registers to be marked as available for allocation.

38. (Currently amended) The system of claim 37, wherein the register allocator to further is capable of, if a preserved register is not available from the list of previously used preserved registers: attempting

attempt to find a tentative live range that includes a second preserved register that is in the list of previously used preserved registers to be allocated to a tentative live range of a second operand.[]] and if successful, allocating

reallocate the second this preserved register for to the operand for its the live range, defined by the operand; and

reserve reserving a different preserved register for to the second operand for its the tentative live range when the attempt to find the second preserved register allocated to the second operand for its tentative live range is successful.

39. (Currently amended) The system of claim 38, wherein the register allocator to further is capable of, if the attempt to find a tentative live range that includes a preserved register that is in the list is not successful: adding

add a new preserved register to the list of previously used preserved registers[]] and allocating this

allocate the new preserved register to the operand for its the live range defined by the operand when the attempt to find the second preserved register allocated to the second operand for its tentative live range fails.

40. (Currently amended) The system of claim 35, whercin the register allocator is capable of, when attempting attempt to allocate [[a]] the scratch register to the operand for its live

range comprises the register allocator to further:

determining if a scratch register is available for allocation; if so, allocating a scratch register for the live range defined by the operand; if not, attempting attempt to steal deallocate a second scratch register from a tentative live range of a second operand live range when the attempt to allocate the scratch register to the operand for its live range fails, marked as tentative; determining if the attempted theft was successful; if so, allocating allocate the stolen second scratch register to the operand for its live range defined by the operand when the second scratch register is successfully deallocated from the second operand for its tentative live range,[[;]] and if not, attempting attempt to allocate [[a]] the preserved register to the operand for its live range when the attempt to deallocate the second scratch register from the second operand for its tentative live range fails.

41. (Currently amended) The system of claim 35, wherein the register allocator to further is capable of, when treating a live range as tentative: if both a scratch register and a preserved register are available for allocation, reserving pre-allocate both the scratch register and the preserved register to the operand for its tentative live range,[[;]]

process one or more additional instructions, allowing another instruction to be processed; and wait waiting for more additional information about the tentative live range.

42. (Currently amended) The system of claim 41, wherein the register allocator to further is capable of, when more information is available: if the last use of the live range has been scheduled,

receive the additional information about the tentative live range, allocating the reserved scratch register to the live range defined by the operand, and freeing

deallocate the reserved preserved register from the operand for its tentative live range when the last use of the operand for its tentative live range has been scheduled,[[;]] and if a function call has been scheduled before the last use of the live range or the reserved scratch register has been stolen, allocating the preserved register to the live range defined by the operand and freeing

deallocate the reserved scratch register from the operand for its tentative live range when the function call is scheduled before the last use of the operand for its tentative live range or when the scratch register is reallocated by any other process.

43. (Currently amended) The system of claim 42, wherein the register allocator to further is capable of, when reserving a preserved register for the tentative live range preferring to reserve attempt to allocate an [[a]] currently unused preserved register that is in the list of previously used preserved registers to the operand for its tentative live range, and before attempt to allocate reserving a new unused preserved register to the operand for the its tentative live range when the attempt to allocate the unused preserved register to the operand for its tentative live range fails.

44. (Currently amended) The system of claim [[43]] 41, wherein the register allocator to further is capable of, when treating a live range as tentative: if a scratch register is not available for allocation, attempting attempt to allocate [[a]] the preserved register to the operand for its tentative live range when the scratch register is determined to be unavailable.

45. (Currently amended) The system of claim 35, the register allocator to further

is capable of: if no proper registers are available for allocation, inserting  
insert a register spill when the attempt to allocate the preserved register and the attempt to  
allocate the scratch register to the operand for its live range both fail,[[;]] and  
either re-attempting re-attempt to allocate the preserved register and the scratch register to the  
operand for its live range, a register, or selecting another select a second instruction to  
schedule.

46. (Currently amended) The system of claim 35, wherein the register allocator further comprises  
includes:

a scratch register allocator to allocate scratch registers to operands for their live ranges when the  
live ranges that do not span a function call;

a preserved register allocator to allocate preserved registers to operands for their live ranges  
when the live ranges that span [[a]] the function call; and

a tentative register allocator to allocate either [[a]] the scratch register or [[a]] the preserved  
register to their respective live ranges when it is unknown that are not immediately  
known whether or not the live ranges they span [[a]] the function call.